THREE NEW COMPUTATIONAL METHODS FOR SOLVING

TWO POINT BOUNDARY VALUE PROBLEMS

Kenneth B. Bley

March 1964

P-2878

ABSTRACT

The methods of Quasilinearization, Dynamic Programming, and Invariant Imbedding are of great practical use in transforming two point boundary value problems into forms that are more readily solved numerically. Each of the three methods is discussed through the use of an example.

THREE NEW COMPUTATIONAL METHODS FOR SOLVING TWO POINT BOUNDARY VALUE PROBLEMS

Kenneth B. Bley*

The RAND Corporation

The desirable goal of optimization often leads to the undesirable requirement for the solution of two point boundary value problems. This paper discusses, through an example, three methods which reduce the two point boundary value problem to a more tractable form suitable for solution on digital computers.

To illustrate these methods we will consider a single cost function which we will seek to optimize (i.e. minimize) using each of the three techniques.

Suppose we wish to find the motion of a particle in an inverse square gravity field. Hamilton's principle states that the motion is given by the function that minimizes the time integral of the Lagrangian where the initial and final states of the particle are given.[1] If an initial state and a time are specified then we have a free boundary problem and the velocity of the particle must be zero at the specified time. We therefore define a cost functional $J[u]$ as

$$J[u] = \int_0^T \left(\frac{1}{2}\dot{u}^2 + \frac{K}{u}\right)dt \qquad (1)$$

where we have, as boundary conditions,

$$\begin{aligned} u(o) &= c \\ \dot{u}(T) &= o \end{aligned} \qquad (2)$$

---

*Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of The RAND Corporation of the official opinion or policy of any of its governmental or private research sponsors. Papers are reproduced by The RAND Corporation as a courtesy to members of its staff.

1. EULER EQUATIONS AND QUASILINEARIZATION

In the preceeding section we defined a cost function which we wish to minimize through the choice of a suitable $u(t)$. If we write

$$J[u] = \int_0^T F(u,\dot{u})dt \tag{3}$$

then Euler's Equation,

$$\frac{\partial F}{\partial u} - \frac{d}{dt}\left(\frac{\partial F}{\partial \dot{u}}\right) = 0 \tag{4}$$

which is a necessary condition for an extremum of the integral, leads to the nonlinear second order differential equation

$$\ddot{u} + \frac{K}{u^2} = 0 \tag{5}$$

We will reduce Eq (5) to two first order equations by taking

$$\begin{aligned} \dot{u} &= v = g(u,v) \\ \dot{v} &= -\frac{K}{u^2} = h(u,v) \end{aligned} \tag{6}$$

subject to the boundary conditions

$$\begin{aligned} u(0) &= c \\ v(T) &= 0 \end{aligned} \tag{7}$$

We are now ready to use the method of Quasilinearization to solve Eq (6). To apply this method we will expand Eq (6) in a Taylor's

series in the functions u and v. If we retain only terms up to first order (as we will do throughout this paper) we will have

$$\dot{u}_{n+1} = g(u_n, v_n) + \left.\frac{\partial g}{\partial u}\right|_{u = u_n} (u_{n+1} - u_n) + \left.\frac{\partial g}{\partial v}\right|_{v = v_n} (v_{n+1} - v_n)$$

$$= v_{n+1}$$

$$\dot{v}_{n+1} = h(u_n, v_n) + \left.\frac{\partial h}{\partial u}\right|_{u = u_n} (u_{n+1} - u_n) + \left.\frac{\partial h}{\partial v}\right|_{v = v_n} (v_{n+1} - v_n)$$

$$= - \frac{3K}{u_n^2} + \frac{2K}{u_n^3} u_{n+1} \tag{8}$$

It is important to notice that Eq (8) are linear in u_{n+1} and v_{n+1}. We will make use of the property of linear equations that the solution is the sum of the particular solution plus the sum of weighted homogeneous solutions.

If we knew $u_n(t)$ and $v_n(t)$ we would be in a position to find $u_{n+1}(t)$ and $v_{n+1}(t)$. We start, then, by assuming a solution which is compatible with the boundary conditions given in Eq (7)

$$u_o(t) = c$$

$$v_o(t) = o \tag{9}$$

Using these functions to evaluate $u_o(t)$ and $v_o(t)$ at every point we will proceed to finding $u_1(t)$ and $v_1(t)$--our new estimate of the solution. Begin by finding the particular solution corresponding

to the initial conditions

$$u_1(o) = o \qquad (10)$$
$$v_1(o) = o$$

Solving Eq (8) subject to these conditions yields

$$u_1^p(t) = q_1(t) \qquad (11)$$
$$v_1^p(t) = q_2(t)$$

We next find the homogeneous solutions by letting all terms involving only u_o and v_o be zero. Solve Eq (8) twice by first letting

$$u_1(o) = 1 \qquad (12)$$
$$v_1(o) = o$$

and second letting

$$u_1(o) = o \qquad (13)$$
$$v_1(o) = 1$$

The initial conditons in Eq (12) give rise to the solutions

$$u_1^{h}1(t) = r_1(t) \qquad (14)$$
$$v_1^{h}1(t) = r_2(t)$$

While the second set of initial conditions, Eq (13), has as its solution

$$u_1^{h2}(t) = s_1(t)$$
$$v_1^{h2}(t) = s_2(t) \tag{15}$$

The total solution is then

$$u_1(t) = q_1(t) + c_1 r_1(t) + c_2 s_1(t)$$
$$v_1(t) = q_2(t) + c_1 r_2(t) + c_2 s_2(t) \tag{16}$$

where the weighting coefficients are determined by using the boundary conditions of Eq (9). We may now continue in a similiar fashion to find $u_2(t)$ and $v_2(t)$, and $u_3(t)$ and $v_3(t)$, etc. until we are satisfied that further computation is unnecessary. Since Quasilinearization has the property of quadratic convergence--each iteration has the effect of virtually doubling the number of correct digits--very few iterations are required.

There are two disadvantages to this method. The first is that solving for the weighting coefficients, involving as it does the solution of linear equations, can lead to errors if the matrix of the homogeneous solutions, evaluated at the requisite points, is ill-conditioned.[2] The second is that the solution to the n+1st solution requires that the entire nth solution be stored in the memory of the computer. This can quickly lead to an unacceptable condition. A way to avoid this[3] is to store only the <u>initial</u> conditions of the 1st, 2nd, ..., nth solution and to solve, at the n+1st iteration, all of the preceeding n solutions simultaneously. We still have a positive gain for we have

traded limited computer storage capability for increased computation--the thing a digital computer does best.

2. DYNAMIC PROGRAMMING

We begin by defining an optimal--a minimum in our case--cost function

$$f(c,T) = \min_u J[u] = \min_u \int_0^T F(u,\dot{u})dt \tag{17}$$

Here $f(c,T)$ represents the minimum possible cost when we start in state c with time T remaining and proceed along the optimum path $u(t)$. It is important to notice that if we have no time $(T = o)$ remaining to our process we can do nothing that will change the value of the incurred cost regardless of the state of the system. That is to say

$$f(c,o) = o \tag{18}$$

The principle of optimality states, "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."[4]

To implement the principle of optimality let us consider the system with $T - \Delta$ remaining. The cost incurred by the system is, to first order, $F(c,v)\Delta$, where here again we have

$$v = \dot{u} \tag{19}$$

No longer are we in state c; rather we now have

$$u = c + v\Delta \tag{20}$$

Applying Eq (17) we can say that the minimum (optimum) cost incurred in a process initially in state $c + v\Delta$ with $T - \Delta$ remaining is $f(c + v\Delta, T - \Delta)$. If we truly wish the optimum cost function we must satisfy the relationship

$$f(c,T) = \min_{v}\left\{F(c,v)\Delta + f(c + v\Delta,\ T - \Delta)\right\} \tag{21}$$

The value of v resulting in the minimization is then our desired v.

Eq (21) can now be expanded in a Taylor's series

$$f(c,T) = \min_{v}\left\{F(c,v)\Delta + f(c,T) + \frac{\partial f}{\partial c}v\Delta - \frac{\partial f}{\partial T}\Delta\right\} \tag{22}$$

$f(c,T)$ appears on both sides and is independent of v so it can be cancelled. We can substitute in for $F(c,v)$

$$F(c,v) = \frac{1}{2}v^2 + \frac{K}{c} \tag{23}$$

Dividing through by Δ we have

$$0 = \min_{v}\left\{\frac{1}{2}v^2 + \frac{K}{c} + \frac{\partial f}{\partial c}v - \frac{\partial f}{\partial T}\right\} \tag{24}$$

In this case we can find the v that will minimize this expression analytically; generally this would involve search techniques on a computer. Here the optimum v is found using basic calculus as

$$v = -\frac{\partial f}{\partial c} \tag{25}$$

Substituting this back into Eq (24) and rearranging leads to

$$\frac{\partial f}{\partial T} = -\frac{1}{2}\left(\frac{\partial f}{\partial c}\right)^2 + \frac{K}{c} \tag{26}$$

Eq (26) can be solved analytically at each point on a grid, with a suitable number of points, in the c,T plane. At each point we would then have both the minimum cost to be incurred, if we carried out the process in an optimal fashion from that point, and the value of v corresponding to the optimal path at that point. In other words we have a feedback process which always indicates the correct direction to proceed in. Starting at any point, u(t) is then completely specified in terms of v(t).

4. INVARIANT IMBEDDING

Again we start with the system of two first order equations given in Eq (6) subject to the boundary conditions given in Eq (7). We wish to imbed our original problem in a more general one valid for any initial position d and any initial time τ. It is obvious that the correct choice of v initially will be a function of initial position and initial time. We note this dependence by saying

$$v(\tau) = r(d,\tau) \tag{27}$$

Now we will consider the process at time $\tau + \Delta$.

$$\begin{aligned} u(\tau+\Delta) &= u(\tau) + v(\tau)\Delta \\ &= d + r(d,\tau)\Delta \end{aligned} \tag{28}$$

$$\begin{aligned} v(\tau+\Delta) = r[d + r(d,\tau)\Delta,\ \tau + \Delta] &= v(\tau) + \dot{v}(\tau)\Delta \\ &= r(d,\tau) - \frac{K}{d^2}\Delta \end{aligned} \tag{29}$$

Expanding the left side of Eq (29) we have

$$r(d,\tau) + \frac{\partial r}{\partial d}\, r(d,\tau)\Delta + \frac{\partial r}{\partial \tau}\Delta = r(d,\tau) - \frac{K}{d^2}\Delta \qquad (30)$$

Simplifying and rearranging yields

$$\frac{\partial r}{\partial \tau} = - \frac{\partial r}{\partial d}\, r(d,\tau) - \frac{K}{d^2} \qquad (31)$$

Eq (31) represents a feedback control law which can be solved for any position d and any time τ subject to the condition

$$r(d,T) = o \qquad (32)$$

In actually implementing Dynamic Programming and Invariant Imbedding digital computers would be used to numerically solve the feedback control laws. The accuracy of the solution then becomes a function of the number of points used in the solution grid and may represent a computer storage problem. Obviously this is not unique to the methods discussed but rather is common to any numerical solution of partial differential equations.

BIBLIOGRAPHY

Since this discussion was meant only to point out the uses of three powerful methods of solving two point boundary value problems no effort was made to maintain mathematical rigor nor was there any discussion of the many and varied uses of these methods. The following bibliography will remedy both these lacks and supply the basic information for those who wish to pursue these methods in more detail.

QUASILINEARIZATION

1. R. Kalaba, "On Nonlinear Differential Equations, the Maximum Operation and monotone Convergence," J. of Math. and Mech., v. 8 (1959), pp. 519-574.

2. R. Bellman, H. Kagiwada, R. Kalaba, "Orbit Determination as a Multi-point Boundary-value Problem and Quasilinearization," Proc. Nat. Acad. Scie USA, v. 48 (1962), pp. 1327-1329.

3. R. Kalaba, "Some Aspects of Quasilinearization," a chapter in the book Nonlinear Differential Equations and Nonlinear Mechanics, Academic Press, N. Y., 1963.

DYNAMIC PROGRAMMING

4. R. Bellman, Adaptive Control Processes, Princeton Univ. Press, Princeton, 1961.

INVARIANT IMBEDDING

5. R. Bellman, R. Kalaba, M. Prestrud, Invariant Imbedding, and Radiative Transfer in Slabs of Finite Thickness, Amer. Elsevier Pub. Co., N. Y., 1963.

6. R. Bellman, R. Kalaba, and G. M. Wing, "Invariant Imbedding and the Reduction of Two-point Boundary Value Problems to Initial Value Problems," Proc. Nat. Acad. Sci. USA, v. 46 (1960), pp. 1646-1649.

7. R. Bellman and R. Kalaba, "On the Fundamental Equations of Invariant Imbedding - I," Proc. Nat. Acad. Sci. USA, v. 47 (1961), pp. 336-338.

ACKNOWLEDGMENT

I would like to thank Dr. R. E. Kalaba for his suggestions and guidance concerning the material presented here.

REFERENCES

1. Weinstock, R., Calculus of Variations, (McGraw-Hill Book Company, Inc., New York, 1952), Ch. 6, p. 74.
2. Bellman, Richard, Robert Kalaba, and JoAnn Lockett, Dynamic Programming and Ill-Conditioned Linear Systems, The RAND Corporation, RM-3815-PR, December, 1963.
3. Kalaba, R. E., Class Notes, Engineering 182D, Variational Methods in Engineering, UCLA, 1963.
4. Bellman, R., Adaptive Control Processes, (Princeton University Press, Princeton, 1961), p. 57.